REMARKS

Favorable reconsideration of this application, in light of the following discussion and in view of the present amendment is respectfully requested.

Claims 1-17 are pending in the present application. Claims 1-12 are amended and claims 13-17 are added by the present Amendment.

I. OBJECTION TO THE SPECIFICATION AND CLAIMS

The specification and claims 2, 4, 8, and 12 were objected to for informalities. The specification and claims 2, 4, 8, and 12 are amended in light of the comments noted in the outstanding Office Action. Accordingly, it is respectfully requested these rejections be withdrawn.

II. REJECTION UNDER 35 USC § 112

Claims 2, 4, 8, 10, and 12 were rejected under 35 USC § 112, ¶ 2 as indefinite. This rejection is respectfully traversed.

Claims 2 and 10 are amended to remove the word "may" in light of the comments noted in the outstanding Office Action. Accordingly, it is respectfully requested that the rejection of claims 2 and 10 be withdrawn.

Further, regarding the phrase "the group" recited in claims 4, 8, and 12, it is respectfully submitted this phrase represents part of a Markush group and is specifically permittable terminology according to the MPEP. For example, MPEP 2173.05(h) recites "one acceptable form of alternative expression, which is commonly referred to as a Markush group, recites members as being 'selected from the group consisting of A, B, and C.' See *Ex parte Markush*, 1925 C.V. 126 (Comm'r Pat. 1925)."

Accordingly, it is respectfully requested the rejection of claims 4, 8, and 12 be withdrawn.

III. REJECTIONS UNDER 35 USC § 103

Claims 1, 2, 4-6, 8-10, and 12 were rejected under 35 USC § 103(a) as unpatentable over pages 1-4 of the originally filed specification (herein "the cited portion") and U.S. Patent No. 4,819,233 to Delucia (herein "Delucia"). This rejection is respectfully traversed.

Amended independent claim 1 recites "disposing a comment statement including a unique comment keyword at all positions in a source code where a comment statement can be disposed," support for which is found in the originally filed specification at least in Fig. 2 and at page 13, lines 8-11.

Amended independent claim 1 further recites "generating specification data by extracting the comment statements each including the corresponding unique comment keyword and joining the comment statements together," and "replacing the comment statement in the source code with the comment statement in the specification data whose corresponding unique comment keyword is coincident with the corresponding unique comment keyword of said comment statement in the source code," support for which is found in the originally specification at least at page 16, lines 7-23. Independent claims 5 and 9 are amended to recite similar features.

As an advantage, in a non-limiting example, comment statements in the source code including comment keywords which are uniquely identified are individually managed by their own comment keywords. Thus, if the comment statements are extracted from the source code to generate specification data and then the specification data is edited by an operator, the changes caused by editing in the specification data are reflected in the comment statements in the source code, with their comment keywords used as keys (see the specification at page 5, lines 6-15). Further, both the first type and second type of comment statements--that is, comment statements written just before an aggregate and comment statements written within an aggregate having processing procedures, respectively--may be managed according to the present invention.

In contrast, the cited portion at page 2, lines 8-18 only discusses extracting the first type of comment statements (that is, comment statements written just before an aggregate) from a source code and editing the extracted first type of comment statements. However, as discussed in the cited portion at page 2, line 27 to page 3, line 15, "the specification generating system according to the prior art explained above does not regard the second type of comment

statement as a processing target." Therefore, it is respectfully submitted the cited portion does not discuss or suggest "disclosing a comment statement including a corresponding unique comment keyword at all positions in a source code where a comment statement can be disposed," as recited in independent claims 1, 5, and 9.

Further, it is respectfully submitted the cited portion only discusses referencing an absolute position of a description line number in a source code to update the source code, but does not discuss or suggest "disposing a comment statement including a corresponding unique comment keyword," as recited in amended independent claims 1, 5 and 9. Rather, because the cited portion only discusses counting line numbers to identify the position of comment statements, it is respectfully submitted the cited portion does not discuss or suggest use of a "unique comment keyword" as recited in the amended independent claims.

Moreover, because the cited portion only discusses editing comment statements of the first type of comment statement, it is respectfully submitted the cited portion does not discuss or suggest "generating specification data by extracting the comment statements each including the corresponding unique comment keyword and joining the comment statements together," as recited in amended independent claims 1, 5 and 9. Rather, the cited portion does not discuss or suggest use of a "unique comment keyword" at all.

In addition, the cited portion at page 1, lines 25 and 26 only discusses updating comment statements of the first type of comment statements. As noted, the cited portion does not discuss use of a unique comment keyword, because the cited portion only refers to editing of the first type of comment statements. Therefore, it is respectfully submitted the cited portion does not discuss or suggest "replacing the comment statement in the source code with a comment code in the specification data whose corresponding unique comment keyword is coincident with the corresponding unique comment keyword of comment statements in the source code," as recited in amended independent claims 1, 5 and 9.

Further, at page 5 of the outstanding Office Action, it is acknowledged that the cited portion does not discuss a unique comment keyword which relates to the source of the specification. Moreover, in the secondary reference, Delucia at col. 2, lines 19-22 only discusses inserting comments with a unique block identifier at the beginning and end of blocks of target units of codes. Delucia at col. 2, lines 48-51 discusses that psuedo code uses the same identifiers as instrumented codes. However, Delucia does not discuss or suggest that the "comment" replaces the "target unit code" after the same "comment" has been edited.

Therefore, it is respectfully submitted that neither Delucia nor the combination of Delucia and the cited portion discuss or suggest "replacing the comment statement in the source code where the comment code in the specification data whose corresponding unique comment key word is coincident with the corresponding unique comment keyword of said comment statement in the source code," as recited in amended independent claims 1, 5 and 9.

Accordingly, it is respectfully submitted amended independent claims 1, 5 and 9 and each of the claims depending therefrom patentably distinguish over the cited portion and Delucia.

Claims 3, 7 and 11 were rejected under 35 U.S.C. § 103(a) as unpatentable over the cited portion, Delucia and Japanese Patent No. 404055938A to Naota (herein "Naota"). This rejection is respectfully traversed.

Claims 3, 7 and 11 depend on amended independent claims 1, 5 and 9, respectively, which as discussed are believed to patentably distinguish over the cited portion and Delucia. Further, Naota only discusses adding keywords to extracted comment statements, and does not discuss or suggest the features of amended independent claims 1, 5 and 9.

Accordingly, it is respectfully submitted independent claims 1, 5 and 9 and claims 3, 7 and 11 depending therefrom also patentably distinguish over the cited portion, Delucia and Naota.

IV. Amendments to Claims, Specification and Drawings

Also, claims 1-12, the specification and FIGS. 2, 7, 9, 11-13, 18 and 22 are amended only to correct minor informalities and to better conform to standard patent practice. It is believed no new matter is added.

V. New Claims

In addition, new claims 13-17 are added to set forth the invention in a varying scope. New independent claim 13 is similar to claim 5, but is directed to a computer. New independent claim 14 also includes similar features to amended independent claim 5, but recites "disposing a comment statement including a corresponding unique comment key word at a plurality of positions within an interior of an aggregate," support for which is found in the originally filed specification at least at page 5, lines 16-23.

Support for new claim 15 is found in the originally filed specification at least at page 10, lines 2-7; support for new claim 16 is found in the originally filed specification at least at page 2, lines 1-3; and support for new claim 17 is found in the originally filed specification at least at page 5, lines 24-26. New claims 15-17 depend on new independent claim 14, and new claims 13-17 are believed to be allowable at least for similar reasons as amended independent claim 5. It is believed no new matter is added.

VI. Conclusion

Consequently, in light of the above discussion and in view of the present amendment, this application is believed to be in condition for allowance and an early and favorable action to that effect is respectfully requested.

If there are any additional fees associated with filing of this Amendment, please charge the same to our Deposit Account No. 19-3935.

Respectfully submitted,

STAAS & HALSEY LLP

Date: July 9, 2004

By: Ryan Rafferty
Ryan Rafferty
Registration No. 55,556

1201 New York Avenue, NW, Suite 700
Washington, D.C. 20005
Telephone: (202) 434-1500
Facsimile: (202) 434-1501

ANNOTATED SHEET

Fig. 2

```
//outline
//    obtain line count of file
//  return value
//    line count of file
//  explanation of parameter
//    nothing
int  getline()
{
   //initialize variable
   int   c, nl;
   nl=0;
   // count number of lines till EOF is detected
   while((c=getchar())!=EOF)
         if(c=='\n')
               ++nl;
   // display line count on screen
   printf("%d\n", nl);
   returun nl;
}   return
```

ANNOTATED SHEET

Fig. 7

```
//outline
//   obtain line count of file
//  return value
//   line count of file
//  explanation of parameter
//   nothing
int  getline()
{
  //initialize ~~variavle~~variable
  int  c, nl;
  nl=0;
  // count number of lines till EOF is detected
  while((c=getchar())!=EOF)
        // check whether it is line feed signal or not
        if(c=='\n')
              // count number of lines in the case of line feed signal
              ++nl;
  // display line count on screen
  printf("%d\n", nl);
  // return line count to caller
  ~~returun~~ nl;
    return
}
```

ANNOTATED SHEET

Fig. 9

```
//outline
//    obtain line count of file
//  return value
//    line count of file
//  explanation of parameter
//    nothing
int  getline()
{
   int   c, nl;
   nl=0;
   while((c=getchar()) !=EOF)
         if (c=='\n')
                ++nl;
   printf("%d\n", nl);
   returun nl;
}   return
```

ANNOTATED SHEET

Fig. 11

```
//outline
//   obtain line count of file
//  return value
//   line count of file
//  explanation of parameter
//   nothing
int  getline()
{
   //1) initialize ~~variavle~~ variable
   int  c, nl;
   nl=0;
   //2) count number of lines till EOF is detected
   while((c=getchar())!=EOF)
         //2.1) check whether it is line feed signal or not
         if(c=='\n')
               //2.1.1) count number of lines in the case of line feed signal
               ++nl;
   //3) display line count on screen
   printf("%d\n", nl);
   //4) return line count to caller
   ~~returun~~ nl;
     return
}
```

ANNOTATED SHEET

Fig. 12

```
//outline
//    obtain line count of file
//  return value
//    line count of file
//  explanation of parameter
//    nothing
int  getline()
{
   //1)
   int   c, nl;
   nl=0;
   //2)
   while((c=getchar()) !=EOF)
         //2.1)
         if (c=='\n')
               //2.1.1)
               ++nl;
   //3)
   printf("%d\n", nl);
   //4)
   ~~returun~~ nl;
    return
}
```

ANNOTATED SHEET

Fig. 13

```
//outline
//   obtain line count of file
//  return value
//   line count of file
// explanation of parameter
//   nothing
int  getline()
{
  //1) initialize ~~variavle~~ variable
  int  c, nl;
  nl=0;
  //2) count number of lines till EOF is detected
  while((c=getchar()) !=EOF)
        //2.1)
        if (c=='\n')
              //2.1.1)
              ++nl;
  //3) display line count on screen
  printf("%d\n", nl);
  //4)
  ~~returun~~ nl;
}  return
```

ANNOTATED SHEET

Fig. 18

```
//outline
//    obtain line count of file
//  return value
//    line count of file
//  explanation of parameter
//    nothing
int  getline()
{
   //1) initialize ~~variavle~~variable
   int  c, nl;
   nl=0;
   //2) count number of lines till EOF is detected
   while((c=getchar()) !=EOF)
         //2.1) check whether it is line feed signal or not
         if(c=='\n')
               //2.1.1) count number of lines in the case of line feed signal
               ++nl;
   //3) display line count on screen
   printf("%d\n", nl);
   //4) return line count to caller
   ~~returun~~ nl;
}   return
```

ANNOTATED SHEET

Fig. 22

```
//outline
//    obtain line count of file
//  return value
//    line count of file
//  explanation of parameter
//    nothing
int  getline()
{
   //2) initialize variable
   int  c, nl;
   nl=0;
   //3) count number of lines till EOF is detected
   while((c=getchar()) !=EOF)
         if (c=='\n')
               ++nl;
   //6) display line count on screen
   printf("%d\n", nl);
   returun nl;
   return
}
```